
Documentation

Passage CMS 2.0 Configuration manual

Date:
Status

January 2006
0.1

Lotus Domino is een handelsmerk van IBM, Verenigde Staten

Careframe, Passage en Notes2SQL zijn handelsmerken van Intraworks, Nederland

Intraworks Solutions B.V.
Bouwerij 81
1185 XW Amstelveen
T +31 (0)20 453 9501

Kamer van Koophandel 34142332
Rabobank 392306182
office@intraworks.nl
www.intraworks.nl

INHOUDSOPGAVE

0	VERSIEBEHEER, DISTRIBUTIELIJST EN PROJECTROLLEN	4
1	MANUALS EXTRANET	5
1.1	INTRODUCTION (PASSAGE CONFIGURATION)	5
1.2	AIM	5
1.3	RELEVANT LITERATURE	5
1.4	THE STRUCTURE OF THIS MANUAL	5
1.5	ADDITIONAL INFORMATION	6
2	PASSAGE CONFIGURATION OVERVIEW	6
2.1	PASSAGE USER-INTERFACE AND PRESENTATION DEFINITION	6
2.2	PASSAGE CONFIGURATION DOCUMENTS	7
2.3	ACCESS TO CONFIGURATION MENU (ADVANCED CONFIGURATION)	8
3	GENERIC ELEMENTS	10
3.1	ADMINISTRATION DOCUMENT	10
3.1.1	<i>Keywords managed on the administration document</i>	<i>12</i>
3.1.2	<i>Library Category definition</i>	<i>12</i>
3.1.3	<i>General Database Graphics</i>	<i>13</i>
3.2	IMAGE STORE	14
3.2.1	<i>What's new view Icons (special Image store document)</i>	<i>15</i>
3.3	KEYWORD LIBRARY	15
3.4	DOCUMENT TEMPLATES	16
3.5	MASTER SLAVE CONFIGURATION	18
4	DEFINING THE SCREEN LAYOUT (FRAMED SITES)	18
4.1	STANDARD FRAME ARCHITECTURE	19
4.2	PASSAGE FRAME DEFINITION	19
4.3	PASSAGE ACTIONS	22
4.4	LAUNCHING PASSAGE FROM THE WEB-BROWSER	23
4.5	ADDITIONAL FRAME DEFINITIONS	23
5	NAVIGATORS	24
5.1	BUTTONS DEFINITION DOCUMENTS	24
5.2	CONTROLLING THE LAYOUT OF THE BUTTON THROUGH A CSS DEFINITION	26
5.3	BUTTON IMPORT / EXPORT FACILITY	27
6	LAYOUT DEFINITION	28
6.1	GENERIC FIELDS ON THE LAYOUT DEFINITION	28
6.2	HTML LAYOUT OF THE PAGE	31
6.3	DETERMINING THE USED LAYOUT DEFINITION	33

7	WEB FRONT END FORMATTING ISSUES	35
7.1	JAVASCRIPT LIBRARY.....	35
7.2	CASCADING STYLE SHEET (CSS) DEFINITION DOCUMENTS	35
7.3	JAVASCRIPT VIEWS.....	36
8	MAIL-IN FILTERS	39
8.1	INTRODUCTION	39
8.2	DEFINING MAIL-IN FILTERS.....	39
8.3	STRUCTURED MAIL IMPORT	40
9	DOCUMENT LIFE CYCLE MANAGEMENT	41
10	CUSTOMISIZING PASSAGE CODE	41
10.1	CONFIGURING THE DATABASE / SECURITY	41

0 Versiebeheer, distributielijst en projectrollen

Tabel 1: Versiegeschiedenis projectvoorstel

Versie	Datum	Auteur	Aard van de wijziging
0.1			Initiele versie voor CMS 2.0

Tabel 2: Distributielijst

Aan	Versie	Datum
Opdrachtgever	1.0	

Tabel 3: Rollen binnen project

Rol	Naam
Opdrachtgever	
Leverancier	
Vertegenwoordiger gebruikers	
Vertegenwoordiger leveranciers	
Projectleider	
Projectborging	

1 Manuals Extranet

1.1 Introduction (Passage Configuration)

Welcome to the passage Configuration Manual for Passage CMS v 2.0.

To generate a full document, press 'voeg samen en print' button on the right hand side of the page.

1.2 Aim

This manual aims to provide a solid knowledge base for the configuration and management of a Passage site. It has been written for all those involved in these activities, like graphic designers, system managers, information brokers and system integrators.

1.3 Relevant Literature

This manual is not intended as an instruction manual for an Intranet's end-users. A customised employee education program will need to be devised for each particular organisation. The manual is not intended for content managers either. Content managers are referred to the end-user manual. The structure of this manual is based on the sequence of operations involved in configuring Passage. We are assuming that Lotus Notes, the Domino server and the Passage Database(s) have been installed. Readers are expected to have basic knowledge about working with the Internet and some basic HTML knowledge.

1.4 The Structure of this Manual

This manual has the following structure:

Chapter 1	General introduction about Passage
Chapter 2	Passage configuration overview
Chapter 3	The site's administrative document. This document is used to predefine a number of the basic features of an Internet or Intranet site, like whether it will be possible to order brochures via the site and all the keywords to be used in the pull-down lists.
Chapter 4	Defining the screen layout
Chapter 5	Navigators/Buttons. How do users navigate Passage?
Chapter 6	Layout: How the content of a Passage

Chapter 7	database is presented on the Web.
Chapter 8	Web front end formatting issues
Chapter 9	Mail-In filters
Chapter 10	Document life cycle management
Chapter 11	Custom fields in Passage
	Database ACL settings

1.5 Additional Information

An alternative source of information is the Passage support website:
<http://passage.intraworks.nl> and Passage end-user and administration manuals.

2 Passage configuration overview

The Passage configuration determines the functionality of the database. Configurable items of the Passage database are:

<i>Item</i>	
<i>Information types</i>	Definition of content streams, keywords, functions, mainly done on the Administration document
<i>User interface, Navigation</i>	Definition of the navigational structure of the site, mainly done in framedefinitions and navigators (button documents)
<i>Presentation (Graphic design)</i>	The presentation of the database to the end users is defined by: <ul style="list-style-type: none"> • Layout definitions • JavaScript • CSS pages • Image store documents
<i>Workflow</i>	Workflow settings are done on the databases access control list (ACL), as well as groups in the NAB
<i>Security settings</i>	Access rights, read/write, create/delete settings are done on the databases access control list (ACL)

2.1 Passage user-interface and presentation definition

Passage CMS has many options on how to present the data to the End user. The basic split can be made between "Frame" and "Frameless"sites.

For "Frame" sites:

The Passage configuration has the following data model:

- Each database has one or more framesets
- Each frameset has up to 4 frame content definitions (red, green, blue and gold)
- Each frame content definition relates to 1 content type (document, view, button navigator, URL)
- Each content type has its own layout definition
- Each layout definition links its own JavaScript and CSS definition documents

For "Frameless" sites:

The Passage configuration has the following data model:

- The screen layout is defined by splitting the Screen in one or more <DIV> tags. The <DIV> tags are positioned using CSS.
- Each content type is connected to a layout definitiontent type (document, view, button navigator, URL)
- Each layout definition contains the <DIV> set, and links to the corresponding CSS Definitions and JavaScript files. Frequently, additional JavaScript Pages are added to create menu effects based on Button & Library documents.

2.2 Passage configuration documents

Necessary documents in a clean (new) Passage database:

<i>ACL</i>	Correct access settings
<i>Administration or General document</i>	1 always necessary, containing database wide category and keyword definitions
<i>Frame definitions</i>	1 definition with the name MAIN must be present (with Frame Sites, in Frameless Sites no frame definitions needs to be present)
<i>Layout definitions</i>	The following layout definitions must be present: <ul style="list-style-type: none">- Default for documents -- Default for navigators -

	<ul style="list-style-type: none"> - Default for views - Library View Mail Search results \$\$Search
<i>JavaScript</i>	<p>JavaScripts are used for effects on the screen, menu's, view formatting, error checking etc. there is a large number of JavaScript functions to be used in Passage sites.</p> <p>Limitations of Javascripts are the poor debugging possibilities, and the fact that all processing is done in the client side. // JavaScripts delivered in the Passage databases or Templates are delivered AS IS. You are obliged to respect and act in accordance to the Copyright statements of previous users.</p>
<i>Text Pages</i>	<p>The following text pages need to be present in any case:</p> <ul style="list-style-type: none"> · 00 (returnpage after mail submission) Thank you, your message has been received · 01 (returnpage after a Response is submitted) Thank you, your response has been received · 02 (returnpage after a Request is submitted) Thank you, your request has been received · 03 (returnpage after a Presentation is uploaded) Thank you, your presentation has been received · AccessDenied (when accessing a page with a non-published status) - This information can not be accessed from the web! · NotAvailable (page is not available) Not available
<i>Master / Slave configuration</i>	<p>In Large Passage Implementations usually the configuration is stored in a central repository database (the Master) and the Content Databases (the Slaves) depend on the master. This decreases the amount of configuration time enormously.</p>

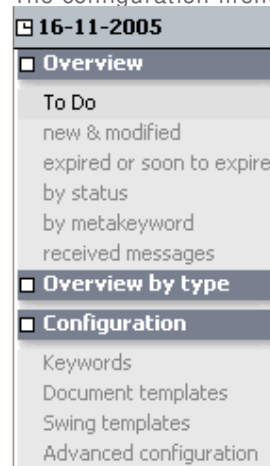
2.3 Access to Configuration Menu (Advanced configuration)

Configuration menu

To change the structure or the appearance of the Intranet or website you need access to the advanced configuration menu.

This access is given to you by the system administrator by setting the right role (**ACLRole: ConfigEd and Access level must be Editor**).

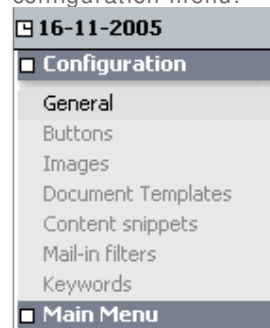
The configuration menu is shown in the bottom left of the menu.



Users obviously require certain rights to access the configuration navigator (**ACLRole: ConfigEd and Access level must be Editor**)

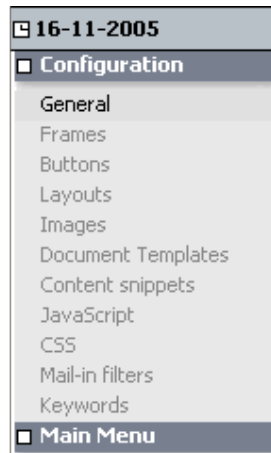
"Slave" configuration menu

A subset of the configuration functions is displayed with the Slave configuration menu:



"Master" configuration menu

all configuration possibilities are display when the master menu is active:



Warning: changing the configuration changes your site. Be ware.
Create backups before doing this.

3 Generic elements

The configuration of the Passage content management system can be divided into 2 parts, generic elements and specific elements.

This chapter describes the generic elements, being:

data-dictionary (administration document)	The data-model of the Passage Content Management system, specifying information types, keywords etc
Image store	Documents containing generic images, to be used anywhere in the site
Keyword library	Documents containing keywords, to be used by custom developments, library documents, meta-information, etc.
Templates	Pre-defined documents that can be pasted into rich text fields, while inheriting styles and layout definitions.
Snippets	Pre-defined pieces of content that can be pasted into rich text fields

3.1 Administration document

The Passage content management system has a very rudimentary datadictionary. The content types in Passage are pre-defined (except for the custom subforms that can be made by very advanced users/developers).

Selecting the Administration document from the Passage advanced configuration menu displays the following Image:

Basics	Keywords - Library documents	Keywords - Other content types	Images	Menu Configuration																		
Basics																						
Database type	Master																					
File names of Passage CMS Slave databases	clients/emergis/portal.nsf																					
Language	English																					
Miscellaneous	<input type="checkbox"/> Allow web editing <input type="checkbox"/> Use !Open instead of ?Open in Button and Framedefinition URLs <input type="checkbox"/> Use Swing for Editors and Editors!C																					
Hide these Workflow elements (Notes client)	<table border="0"> <tr> <td><input type="checkbox"/> Include in search</td> <td><input type="checkbox"/> Action: Close</td> <td><input type="checkbox"/> Action: Publish Immediately</td> </tr> <tr> <td><input type="checkbox"/> Include in logging</td> <td><input type="checkbox"/> Action: Hide status</td> <td><input type="checkbox"/> Action: Archive</td> </tr> <tr> <td><input type="checkbox"/> Icons</td> <td><input type="checkbox"/> Action: Concept</td> <td><input type="checkbox"/> Action: Create Work Copy</td> </tr> <tr> <td><input type="checkbox"/> Metakeywords</td> <td><input type="checkbox"/> Action: Reassign</td> <td><input type="checkbox"/> Action: Reset Fonts</td> </tr> <tr> <td><input type="checkbox"/> Previous & Next buttons</td> <td><input type="checkbox"/> Action: Quality check</td> <td></td> </tr> <tr> <td><input type="checkbox"/> Comments</td> <td><input type="checkbox"/> Action: Publish</td> <td></td> </tr> </table>				<input type="checkbox"/> Include in search	<input type="checkbox"/> Action: Close	<input type="checkbox"/> Action: Publish Immediately	<input type="checkbox"/> Include in logging	<input type="checkbox"/> Action: Hide status	<input type="checkbox"/> Action: Archive	<input type="checkbox"/> Icons	<input type="checkbox"/> Action: Concept	<input type="checkbox"/> Action: Create Work Copy	<input type="checkbox"/> Metakeywords	<input type="checkbox"/> Action: Reassign	<input type="checkbox"/> Action: Reset Fonts	<input type="checkbox"/> Previous & Next buttons	<input type="checkbox"/> Action: Quality check		<input type="checkbox"/> Comments	<input type="checkbox"/> Action: Publish	
<input type="checkbox"/> Include in search	<input type="checkbox"/> Action: Close	<input type="checkbox"/> Action: Publish Immediately																				
<input type="checkbox"/> Include in logging	<input type="checkbox"/> Action: Hide status	<input type="checkbox"/> Action: Archive																				
<input type="checkbox"/> Icons	<input type="checkbox"/> Action: Concept	<input type="checkbox"/> Action: Create Work Copy																				
<input type="checkbox"/> Metakeywords	<input type="checkbox"/> Action: Reassign	<input type="checkbox"/> Action: Reset Fonts																				
<input type="checkbox"/> Previous & Next buttons	<input type="checkbox"/> Action: Quality check																					
<input type="checkbox"/> Comments	<input type="checkbox"/> Action: Publish																					
E-mail address(es) to send notifications to about expired documents and documents that need to be revised																						

Modification logging

Modified by	Date
Robert Lukkenaer/Intraworks	15-11-2005 15:10:25
Jasper Goertz/Intraworks	09-11-2005 09:57:37
Jasper Goertz/Intraworks	09-11-2005 09:57:30

The data-dictionary is managed via the Administration document, this document contains:

- keywords per data-type
- generic icons
- when applicable, mail-adresses for users handling specific events (mail-forms)
- database language, enable web-editing
- Master / Slave setting
- Hide workflow elements

A special data-type in Passage are the library documents. The configurator can define a number of information categories (this used to be 31, but this limitation is removed, do not use more than 50 library categories for realistic usage).

For each information category the configurator can specify a set of keywords (keyword 1 & 2). This function is present for backward compatibility, It is better to use Keyword documents

all functions on the Administration document are "self explanatory" and when

applicable accompanied by an Helper tekst.

3.1.1 Keywords managed on the administration document

Each information type (specified below), except for the library documents, has specific fields that determine the functionality. These fields are:

- keywords
- mail-adresses
- some keyword dependent types

These keywords are managed on the administration document.

Basics	Keywords - Library documents	Keywords - Other content types	Images	Menu Configuration
Address book				
Departments	aap jet mies noot pijp schaap vliegtuig			
Brochures				
Keywords	Used to categorize Brochures	aap jet mies noot pijp schaap vliegtuig		
Orderable	Can Brochures be ordered? External - requires users to fill in company name, address etc. Internal - only requires the users' name to be filled in	<input type="radio"/> Yes - External <input checked="" type="radio"/> No <input type="radio"/> Yes - Internal		
Editor	Next editor after a Brochure has been ordered from the web			
Show number and publishing date on the web?		<input type="radio"/> Yes <input checked="" type="radio"/> No		

3.1.2 Library Category definition

On the Administration document the library categories can be defined. In versions prior to 1.82 this was limited to 31, current versions do not have this user limit.

Basics	Keywords - Library documents	Keywords - Other content types	Images	Menu Configuration
Keywords for Library documents				
Information category names (the maximum number of entries is 31)	The numbers to the left of each category indicate which view is associated with that category.	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	<p>Be careful changing the value of this field!</p> <p>Do not delete a value in the middle of the list; if you do not need a value anymore, simply replace it with another value. If a value is deleted, all documents that use the consequent values will not have the proper categories assigned to them anymore!</p> <p>It is always possible to append values.</p>	
Keywords 1 Keywords 2				

The number of specific views (by name / by keyword) remains limited to 31.

!!! Do NOT renumber !!!

Library categories are defined by number and by name. When defining library categories, it is very IMPORTANT, that you do not change or renumber the library categories, since the views use a name/number conversion to select documents (done for performance reasons). Renumbering the library definitions without taking precautions will definitely result into jibberish.

3.1.3 General Database Graphics

There are still some graphics that are managed on the administration document. Please refer to the following image.





Basics	Keywords - Library documents	Keywords - Other content types	Images	Menu Configuration
--------	------------------------------	--------------------------------	---------------	--------------------


General database graphics

This field must contain the following pictures:




Navigation:	Vacancies:
COLLAPSE.GIF	REAGEER.GIF
EXPAND.GIF	REAGEERMANNETJE.GIF
NEXT.GIF	
PREVIOUS.GIF	
SEARCH.GIF	
REAGEER2.GIF	

Navigation: used in views and forms; **Vacancies:** used in Vacancies.
 If you decide to replace one of the existing pictures, make sure that the name of each new picture is the same as the one you want to replace. On Linux machines, these names are case-sensitive!



Essential graphics:

Backgrounds

This field contains all pictures that can be used as backgrounds (in Layout Definitions).

3.2 Image store






The image store is used to archive all graphic elements except for the General Database Graphics. The images are stored in several image store documents identified by name.

for example 'drempels weg' functionality images are stored in this image store:

Basics

Basics	
Document key	drempelsweg
Document URL	/clients/emergis/master.nsf/v9906/drempelsweg/\$File/'attachment name'
Description	Drempelsweg

Attachments

Attachments	
	zetopzw.gif
	Font-small.gif
	zetop.gif
	Font-large.gif
	Font-medium.gif
	zetoprgb.gif

3.2.1 What's new view Icons (special Image store document)

There is a special what's new view that allows adding images or icons to the documents shown in the what's new view. On each document it is possible to select the images from a special image store document with the following key: **PASSAGE_ICONS** This document is a regular image store document but has a reserved name to it.

3.3 Keyword Library

The keyword library is used for storing extra sets of keywords. This can be useful for customising Passage.

Meta keywords for the sitemap functionality and relevant documents are also stored here.

Meta keywords may not contain any spaces. Replace a space by an underscore.

Basics

Basics

Keyword group	PASSAGE LIBRARYKEYWORDS 1
Keyword name	KE1
Description	First level of keywords....
Resulting key	PASSAGE LIBRARYKEYWORDS 1_KE1 The keywords in this list will be available for Library Category "KE1"
Lookup code	sLookup_Key := "PASSAGE LIBRARYKEYWORDS 1_KE1"; sLookup_Temp := @DbLookup("" : "NoCache"; "" : "v9916BBS1"; sLookup_Key; 2); @if(@IsError(sLookup_Temp); "" : sLookup_Temp)

Keywords

Keywords

aap
mies
noot

Sort

3.4 Document Templates

Using a document template is the only way documents can be created in Passage CMS (using the Lotus Notes client). Templates are pre-defined documents that guide the editors to correct output that guarantees performance and consistency.

For configurators: Copy all templates from the Passage CMS template.

You can create your own template using the template functionality. Templates are synchronized with the Passage configuration at creation time.

Basics	Custom fields
Basics	
Description for this document template	MP3
Document type	Library

General	Categorization	Other
General		
Title	<Please enter a title>	
Status	Published	
Summary	Tesaamvatting	
Comments	Commentaar	
Author		
Source		

Using the custom fields functionality, you can create your own, advanced templates, using evaluated @formulas.

Basics	Custom fields
Custom fields	
Field name	@Formula for field value
Title	@UserName + @Text(@Now)

General	Categorization	Other
General		
Title	<Please enter a title>	
Status	Published	
Summary	Tesaamvatting	
Comments	Commentaar	
Author		
Source		

3.5 Master Slave configuration

In the Master Slave configuration (from Passage 1.83) multiple Passage databases are specially configured such that they share configuration and design elements. The following table shows the difference between Master and Slave.

Master	Slave
Administration document	Administration document
Buttons	Buttons
Images	Images
Templates	Templates
Mail-In filters	Mail-In filters
Keywords	Keywords
Framesets	
Layouts	
JavaScripts	
CSS	

Master Slave configuration is controlled on the Database Administration Document, just set the DB to slave, and specify the Master name and the system is functional.

The screenshot shows a web-based form titled "Database administration - Manuals Extranet". At the top, there is a tab bar with "Basics" selected, followed by "Keywords - Library documents", "Keywords - Other content types", and "Images". Below the tabs is a table with the following fields:

Database type	Slave
File name of the Passage CMS Master database	intraworks\internet\frameless.nsf
Language	English
Miscellaneous	<input checked="" type="checkbox"/> Allow web editing

Pitfalls:

When using JS views the routines need to be changed to specify the Slave database (this is done by changing the variable DBNameHTML into DBNameHTML_C). Only one type of frameset out of all framesets can be used.

Advantages are:

Easy configuration and modification
Speed since fewer documents are used (chaching is optimised).

Available from: Passage 1.83 ML

Top of your administartion document where you have to select if you want a master or a slave database.

4 Defining the screen layout (framed sites)

4.1 Standard Frame Architecture

Passage uses a maximum of 4 frames to structure the website. The Passage frame architecture organises navigation elements hierarchically. There are three different navigation elements:

1. Top navigators (comparable with chapters)
2. Cluster navigators (comparable with paragraphs)
3. Extra or generic Buttons (always available, for example: 'search', 'what's new' and 'mail').


The fourth frame in Passage is being used to publish the content.

New from version 2.0 is the possibility to inject HTML Code Pre- and After the Frameset, thus giving the opportunity to create framesets around the frameset (e.g. for centered sites).

4.2 Passage Frame definition

Any number of framedefinitions can be defined in the Passage database. Normally one or 2 framedefinitions are sufficient.

In a Framed Passage site there must be one frame definition with the name: **MAIN**. This framedefinition is used in the buttons.

Basics	
Basics	
Adressing	MAIN
Description	Homepage
Frameset layout	Alternative 9 

Frame title and Meta Keywords:

Site title and META tags	
HTML Title	Intraworks
META description	Welkom bij Intraworks!
META keywords	Content Management, Intranet, Internet, Lotus Notes, Lotus Domino, Lotus, Notes, Domino, groupware, Passage, Projects, ASP, Consultancy, Consultants, Kennisoverdracht, Nieuwe Media, Nieuwe, Media Communicatie, Samenwerken, Amsterdam, Nederland, Training, Coaching, Advies, CLP, intranet, lotus domino, lotus notes, notes 5.0, notes 4.6, notes r5, business suite, development environment, itf software development kit, itfsdk, itf sdk, itf business partners, lotus business partners, itf.community, itfcommunity, itf community members, itf development tools, buy software tools, buy software solutions, colabration, colaborative, collaborative ebusiness solutions, collaborative e-business solutions, collaborative ebusiness applications, collaborative e-business applications, collaborative business solutions, business to business collaboration, business-to-business collaboration, b2b, itf business suite, itf architected applications, itf-architected applications, lsx, business components, developer tools, business software shop, developer component e-shop, developer component store, component-based development, component based development, rapid application development, add-ons, add-ins, reusable components, re-usable, reuseable, freeware, shareware, api, application programming interface, itf isv, itf asp, lotus isv, lotusphere, workflow, e-crm, The foundation of E-business

Defining the Frame contents can be done using the color identification. The element top open is helped.

Frame contents		Advanced	
1. Red frame (frame name "boven")			
Show a navigator			
Database	- Current -		
Navigator	000000_Intraworks		
Size	110	Resizable	Yes
Border	No width: 0	Margin	width: 0 height: 0
Frame spacing	0	Scrolling	No
2. Green frame (frame name "links")			
Show a document			
Database	- Current -		
View	Text Pages		
Document	Home1		
Size	189	Resizable	No
Border	No width: 0	Margin	width: 0 height: 0
<reset all borders>			
Frame spacing	0	Scrolling	No
3. Blue frame (frame name: "rechts")			
Show a document			
Database	- Current -		
View	Text Pages		
Document	Home2		
Border	No width: 1	Margin	width: 5 height: 0
Resizable	No	Scrolling	Auto
4. Golden frame (frame name: "londer")			
Show a document			
Database	- Current -		
View	Text Pages		
Document	TT Pages		

HTML Code injector:

HTML Code can be injected in the Frame set. This creates many, many possibilities, but you need to have HTML Knowledge.

Frame contents	Advanced
HTML code before the FRAMESET code	
<pre><FRAMESET FRAMEBORDER=no BORDER=0 FRAMESPACING=0 COLS="*,800,*"> <FRAME NAME="BLeft" src="/passage/partner/iw/iw_web.nsf/EMPTY?OpenPage"></pre>	
HTML code after the FRAMESET code	
<pre><FRAME NAME="BRight" src="/passage/partner/iw/iw_web.nsf/EMPTY?OpenPage"> </FRAMESET></pre>	

Google Compatibility

The Frameset generates code that is accepted by Google (nov 2005) for indexation. this feature can change obviously.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<SCRIPT language="JavaScript1.2"
src="/passage/partner/iw/iw_web.nsf/PassageJSLibrary.js?OpenPage"
type="text/javascript"></SCRIPT>
<SCRIPT language="JavaScript" type="text/javascript">
DBNameHTML="/passage/partner/iw/iw_web.nsf";
DBNameHTML_C="/passage/partner/iw/iw_web.nsf";
</SCRIPT>
<!--
Created using Passage CMS
-->
<TITLE> Intraworks</TITLE>
<META NAME="description" content="Welkom bij Intraworks!">
<META NAME="keywords" content="Content Management, Passage Projects, Intranet, Internet,
Lotus Notes, Lotus Domino, Lotus, Notes, Domino, groupware, Passage, Projects, ASP,
Consultancy, Consultants, Kennisoverdracht, Nieuwe Media, Nieuwe, Media Communicatie,
Samenwerken, Amsterdam, Nederland, Training, Coaching, Advies, CLP">
</HEAD>
<FRAMESET FRAMEBORDER=no BORDER=0 FRAMESPACING=0 COLS="*,768,*">
<FRAME FRAMEBORDER=0 SCROLLING=No MARGINWIDTH=0 MARGINHEIGHT=0 NORESIZE
NAME="BLeft" SRC="/passage/partner/iw/iw_web.nsf/v01013/TT_Blank!open">
<FRAME SCROLLING=No MARGINWIDTH=0 MARGINHEIGHT=0 NORESIZE NAME="BContent"
SRC="/passage/partner/iw/iw_web.nsf/framedefinitions/main!OpenDocument">
<FRAME FRAMEBORDER=0 SCROLLING=No MARGINWIDTH=0 MARGINHEIGHT=0 NORESIZE
NAME="BRight" SRC="/passage/partner/iw/iw_web.nsf/v01013/TT_Blank!open">
</FRAMESET>
<A HREF="/passage/partner/iw/iw_web.nsf/v14013!openview">All documents for
crawler</A>
</HTML>

<!--</SCRIPT>
<SCRIPT LANGUAGE="JavaScript1.2"
src="/passage/partner/iw/iw_web.nsf/p0002JVS3?OpenPage"></SCRIPT>

```

4.3 Passage actions

Passage Actions are executed from within a frame or button document. Each action performs a specific action on a notes/passage database or onto another system. Actions are:

Show document	Shows a document from a view. Opens a document from the specified database that is located in the specified view. In the second row you now need to enter which database and view the document comes from.
Create document	Creates a specific document in a Notes database
Show a view	Shows a view from a notes database. View option: Expand (Feature that expands any addressed views) and collapse (Feature that collapses any

	addressed views)
Show a navigator	Shows a button navigator (within Passage)
Open a database	Opens a database with the default database launch options
URL	Anything that can be addressed from the client's system
Open the users own mail file	Opens the users own notes web mail file (only in Passage Intranet version!)

The Passage program handles the correct translation of the Above actions into HTML commands.

4.4 Launching Passage from the web-browser

The Passage database needs to be configured to perform the right action when the database is opened from the web-browser. The only correct action for Passage used to be to open the MAIN frame definition when the database is launched.

Nowadays this is no longer necessary. Passage can open any content element and position a frameset (if required) around the content.

To perform the initial action:

This is configured as follows:

1. Click copy Doclink to mainframe definition to the clipboard from the frame definitions menu
2. Select the database properties box, and select the launch tab
3. From the database launch option, select launch first Doclink in "About database". The database will now launch the mainframe definition when the database is opened from the web.

p.s. You do need designer access to perform this operation.

4.5 Additional Frame Definitions

Intraworks Solutions can develop additional frame layouts on request. Please contact your representative for details.

5 Navigators

Passage navigators are used to navigate a Passage site. Navigators establish the hyperlinks required to access new navigators and other functionality.

Within Passage, navigators are created by grouping button documents by keyword (buttons are explained in the following paragraphs). Using these keywords, passage navigators can be addressed from Frame definitions or from other buttons documents. The following picture shows the button selection and the representation on our website.

The screenshot shows the Passage configuration interface. On the left, there is a sidebar with a tree view containing 'Configuration' (expanded) and 'Main Menu'. Under 'Configuration', there are sub-items: General, Buttons, Images, Templates, Mail-in filters, Keywords, and Main Menu. The 'Main Menu' item is selected. The main area displays a table of buttons. At the top, there is a header bar with the text 'Homepage on Vesuvius/Intraworks' and a date '16-11-2005'. Below the header, there are several checkboxes: 'New Button', 'Refresh All', 'Show Selected', 'Don't Show Selected', and 'Show Document'. The table has three columns: 'Cluster', 'Position', and 'Title'. The table is filtered to show 'Frameless' buttons. The buttons listed are:

Cluster	Position	Title
✓ 05000000.		Home
✓ 1000000.		Het Bedrijf
✓ 1000000.1010000.		Het Bedrijf\Inleiding
✓ 1000000.1070000.		Het Bedrijf\Klanten
✓ 1000000.7010000.		Het Bedrijf\Persberichten
✓ 1000000.7020000.		Het Bedrijf\Nieuwsbrieven
✓ 1000000.900800.		Het Bedrijf\Contact
✓ 2000000.		Producten
✓ 2000000.2010000.		Producten\Inleiding
✓ 2000000.2010500.		Producten\Passage™ CMS
✓ 2000000.2040000.		Producten\Intelligent Search Agent
✓ 2000000.2040010.		Producten\Import Module
✓ 2000000.2040010.		Producten\Passage DNS

5.1 Buttons definition documents

Buttons are a very flexible way of creating a navigation structure in your I-net. Passage Projects is continuously developing new methods to increase the usage of buttons. Please contact Passage Projects on a regular basis for new developments. On each button you can define the following variables:

- Positioning
- Appearance
- Actions
- Audience (in Passage intranet version only)

On each button you can specify:

Basics:

Basics	
Cluster	frameless
Parent	<input type="text"/>
Position	<input type="text" value="05000000"/>

Cluster: The cluster containing the button. Clusters group buttons
Parent: It is possible to create a hierarchy from buttons for easy positioning
Position: The ordering of the button. Various sequences can be created by manipulating the values.

SHOW YES/NO:

SHOW YES/NO:	
Show button	<input checked="" type="radio"/> Yes <input type="radio"/> No
Version	<input type="checkbox"/> V2 Button

Must this button be shown

LAYOUT

Layout		
Type	Orientation	Spacing
<input type="text" value="Text"/>	Horizontal	Horizontal: 0 Vertical: 0
Text Properties		
CSS: class name		
Buttons		
Button Text		
Home		

The presentation of the button. the CSS to be used, if it is a graphic button or not etc. Intraworks recommends: Text buttons with CSS Positioning. This offers maximum flexibility.

AUDIENCE

Audience
People, Groups or Roles that can see this button (none allows all):
<input type="text"/>

The audience option is part of the Intranet version of Passage. This option sets the

readers field of the button. If the reader is in the specified group or role (listed in the Server's central Name and Address book), the button is visible from the Web. Otherwise the button is hidden. In this way the user is offered relevant information only. For more information, please visit the website: www.passage.nl

ACTIONS

Actions

1.

2. Target

Database	- Current -
View	Text Pages <input checked="" type="radio"/> Collapsed <input type="radio"/> Expanded
Document	Index

The action to be performed on the button. This is identical to the actions in the Frames.

Please note that if you use buttons to feed menu's, for example TIGRA menu's, you need a JavaScript view and connector. Only the first action is exported in the JavaScript view.

5.2 Controlling the layout of the button through a CSS definition

It is possible to define special graphical effects on buttons without the creation of special GIF images for each state (for example mouse-over, mouse-down and steady state).

The major advantage of this technique is the speed (no downloads, rendering by the web-browser). A class is selected from the button document:

- Defining an CSS Style that governs the display of the buttons (for example .buttons)
- Creation of a CSS definition page in the Passage configuration menu
- Linking the CSS definition page onto the button navigator using the layout definition

Frequently used example:

- Define a graphics + text buttons, using a mouse- over effect (the image is a generic button image)
- The Cascading Style positions the text on top of the button, for example using the following css definition:

```
.Buttons
{ position: relative;
left:-100px;
top:-17px;
font-family:arial,verdana;
font-weight:bold;
font-size:10pt;
color:# 1D3854
}
```

Where the text normally is positioned at the right-bottom of the image, the relative positioning places the text over the image. For more examples, please refer to the configuration examples on our website. **Even Better: Get a CSS book or browse the Web. CSS Examples are everywhere.**

5.3 Button import / export facility

this function no longer works in Passage CMS.

6 Layout definition

Passage uses Layout definitions to determine how various elements (navigators, documents and views) are presented.

Passage stores content and 'web presentation of this content' in separate locations. Content is stored in various document types, the lay-out is stored in the Configuration menu.

Passage Layout definitions are used to define the layout for the web-browser. When a navigator, document or view is accessed on the web, the layout definition is added to the document. Because content and presentation are stored independently it is easy to change them.

6.1 Generic Fields on the Layout Definition

Layout definitions are the Glue between the content, the navigation and the presentation. there are many ways to create layout definitions. Please be reluctant in building new layout definitions, there must be a smarter way to reach your goal then to maintain 100 layout definitions for a single site.

Identification and Bind (The name of the form or view to which the definition applies.)

Basics	
Form/View	- Default for documents -, Library, Search

HTML Layout of the Page (HTML Head)

It is possible to inject HTML code in the HTML Head of the Page.

HTML layout of the page	JavaScript, CSS and other properties	Link colors
HTML Head - JavaScript code		
<pre><SCRIPT language="JavaScript"> // Load the scripts necessary for Tigr Menu Gold (!!!! should be the last statement in this field !!!!) loadTigrMenuGoldScripts(); // alert (DBNameHTML_C); </SCRIPT></pre>		

HTML Layout of the Page (HTML before the title)

It is possible to inject HTML code in the HTML Page prior to the Title. In this example we use an Frameless site with DrempelsWeg (disability) implementation.

1. < HTML code before the title	(available JavaScript functions)
---	----------------------------------

```

<DIV id="crawlview"><BR><A HREF="/page/homepage.nsf/v14013openview"></A></DIV>
<DIV id="TOP">
  <DIV id="accessibility-css">
    <SPAN id="plain" onclick="CSSCookies.fullreset()"></SPAN>
    <SPAN id="larger" onclick="CSSCookies.setLarger()"></SPAN>
    <SPAN id="largest" onclick="CSSCookies.setLargest()"></SPAN>
    <SPAN id="bw" onclick="CSSCookies.setBW()"></SPAN>
  </DIV>
  <DIV id="searchBox">
    <FORM name="searchform"><NOBR><INPUT name="Search" type="text" class="inputFld" value="Zoeken..." onFocus="this.value=""" onBlur="checkContent(this)" onKeyDown="checkEnter()" onCh
    types="button" value="GAAN" class="inputBtn" style="height:20px;" onclick="openSearch(document.forms[0].Search.value)"></NOBR></FORM> </DIV>
  </DIV>
  <DIV id="MIDDLE">
    <DIV id="MIDDLE-Navigation">
      <!-- Tigra Menu Gold, should be placed just before the </BODY> tag at the end of the document -->
      <SCRIPT language="JavaScript" type="text/javascript">
        showTigraMenuGold( "MENU_ITEMS0", "MENU_SCOPE0", false );
      </SCRIPT>
      <!-- / Tigra Menu Gold -->
    </DIV>
  </DIV>
  <DIV id="BOTTOM">
    <DIV id="BOTTOM_Content"><IMG align="left" height="600" width="1px" src="/intra/works/internet/frameless.nsf/v9906/IW/$File/Vertical.gif" border="0">
    <DIV id="ToPrint">
      <SPAN class="Bottom_Button_Back_Top" onclick="history.go(-1)"><IMG src="/intra/works/internet/frameless.nsf/v9906/IW/$File/Back.gif" border=0 alt="Back"></SPAN>
      <SPAN class="Bottom_Button_Print_Top" onclick="openPrintWindow()"><IMG src="/intra/works/internet/frameless.nsf/v9906/IW/$File/Print.gif" border=0 alt="Print"></SPAN><BR><BR>
    </DIV>
    <DIV id="Contact_Links" class="Links"></DIV>
  </DIV>

```

HTML Layout of the Page (HTML after the title)

After the title but before the notes rich text item containing all content

2. > HTML code after the title	(available JavaScript functions)
--	----------------------------------

```

<BR><BR>

```

HTML Layout of the Page (HTML at the end of the document)

Closing statements at the end of the page

3. - HTML code at the end of the page	(available JavaScript functions)
--	----------------------------------

```

</DIV>
</DIV>

<SPAN class="Bottom_Button_Back" onclick="history.go(-1)"><IMG src="/intra/works/internet/frameless.nsf/v9906/IW/$File/Back.gif" border=0 alt="Back"></SPAN>
<SPAN id="CopyrightNotice" style="font-size: 30%; text-align: center;">&copy; 1996 - 2005 Intra/works Solutions B.V.</SPAN>
<SPAN class="Bottom_Button_Print" onclick="openPrintWindow()"><IMG src="/intra/works/internet/frameless.nsf/v9906/IW/$File/Print.gif" border=0 alt="Print"></SPAN><BR><BR>
</DIV>

```

CSS Files

CSS Files to be bound to this page can be defined on this part. CSS Files are stored in the CSS Store in the Configuration.

CSS	
Standard	
CSS	Default [] TigraMenuGold.css [v1.00 (2005-01-05 RHK) - CSS for Tigra Menu Gold - Based on 'frames' example]
Accessibility (optional)	
Text size: larger	larger [CSS for better visual accessibility - larger fonts font-size:: 95%]
Text size: largest	largest [CSS for better visual accessibility - largest fonts font-size:: 115%]
Color: black & white	bw [Black & White]
HTML example for accessibility buttons:	
<pre><DIV id="accessibility-css"> Normal Larger Largest B&W </DIV></pre>	

JavaScript Files

JavaScript Files to be bound to this page can be defined on this part. JavaScript Files are stored in the JavaScript Store in the Configuration.

JavaScript	
JavaScript	ContentScripts Metakeywords-Formatter Print ShowDocInfo TigraMenuGold_PassageInterface
Make document field values available as JavaScript variables	<input checked="" type="radio"/> Yes <input type="radio"/> No

Document information is available in JavaScript

Specific document information can be made available to the end-user (i.e. browsing the website). This is done by optionally making the document info (or meta data) available to JavaScript. Via a specific function, this info can be placed everywhere on the screen. Information available is:

```
var title = "05.1 Buttons definition documents";
var summary = "";
var author = "Oscar Nijst";
var libraryCategory = "Manuals";
var keywords = "Passage Configuration-CMS 2.0";
var sKeywords_1 = "Passage Configuration";
var sKeywords_2 = "CMS 2.0";
var sKeywords_Meta_1 = "Manuals_Passage_Configuration_2_0";
var sKeywords_Meta_2 = "";
var source = "";
var dateRevision = "";
var dateCreated = "16-11-2005 14:40:56";
var dateModified = "16-11-2005 16:31:00";
var datePublished = "14-05-2002";
var sEmail = "";
var sRefID = "";
var sDocIcons = "";
```

No longer active used are:

Background images

Background image ?	
Image Store document	
Background image	No background

Link colors

HTML layout of the page JavaScript, CSS and other properties Link colors		
NOTE: The following settings are obsolete; use CSS to implement the color for the various link states.		
Link type	Color group	Color
<u>Active link</u> ?	Yellow	DarkGoldenrod
<u>Visited link</u> ?	Blue	blue4
<u>Link</u> ?	Gray	gray6

6.2 HTML layout of the page

As explained, Passage facilitates adding HTML to the presentation of the page on the web. This is done by injecting HTML on the Page. Notes and Passage don't change the HTML in any way. In addition, already existing HTML code from existing layout definitions can be integrated in new definitions.

Passage documents are always presented in the browser using two fields, the **Title** and the **Body**. In the Layout Definition HTML code can be inserted in front of the title, between the title and the body and behind the body field. The Webserver generates the desired screen layout.

An example of how HTML code can be used to determine the layout of a document.

HTML Head

Allow passthrough html to be placed in the html-head part of the HTML document. This is necessary for specific JavaScript functions to work.

Also a generic variable called DBNameHTML (DBNameHTML_C in Master/Slave configurations) is made public on this page.

Necessary for:

- Dynamic HTML
- JavaScript Views
- Browser specific issues (by putting self modifying code in this field)

<SCRIPT language="JavaScript">

```
// Load the scripts necessary for Tigra Menu Gold (!!!! should be the last
statement in this field !!!!)
loadTigraMenuGoldScripts();
// alert (DBNameHTML_C);
```

</SCRIPT>

Before the title

This field can be used to define a table with three columns, for example, placing the next and previous buttons in the first cell, and to defining the layout of the title. The roll down menu (click available JavaScript functions) shows the available JavaScript functions .

```
<DIV id="crawlView"><BR><A
HREF="/passage/homepage.nsf/v14013!openview">.</A></DIV>
<DIV id="TOP">
<DIV id="accessibility-css">
<SPAN id="plain" onclick="CSSCookies.fullreset()"></SPAN>
<SPAN id="larger" onclick="CSSCookies.setLarger()"></SPAN>
<SPAN id="largest" onclick="CSSCookies.setLargest()"></SPAN>
<SPAN id="bw" onclick="CSSCookies.setBW()"></SPAN>
</DIV>
<DIV id="searchBox">
<FORM name="searchform"><NOBR><INPUT name="Search" type="text"
class="InputFld" value="Zoeken..." onFocus="this.value="
onBlur="checkContent(this)" onKeyDown="checkEnter()"
onChange="openSearch(this.value)"> <INPUT type="button" value="GAAN"
class="InputBttn" style="height:20px;"
onclick="openSearch(document.forms[0].Search.value)"></NOBR></FORM>
</DIV>
</DIV>
```



```

<DIV id="MIDDLE">
<DIV id="MIDDLE-Navigation">
<!-- Tigr Menu Gold, should be placed just before the </BODY> tag at the
end of the document -->
<SCRIPT language="JavaScript" type="text/javascript">
showTigrMenuGold( "MENU_ITEMS0", "MENU_SCOPE0", false );
</SCRIPT>
<!-- / Tigr Menu Gold -->
</DIV>
</DIV>
<DIV id="BOTTOM">
<DIV id="BOTTOM_Content"><IMG align="left" height="600" width="1px"
src="/intraworks/internet/frameless.nsf/v9906/IW/$File/Vertical.gif"
border="0">
<DIV id="ToPrint">
<SPAN class="Bottom_Button_Back_Top" onclick="history.go(-1)"><IMG
src="/intraworks/internet/frameless.nsf/v9906/IW/$File/Back.gif" border=0
alt="<< Back"></SPAN>
<SPAN class="Bottom_Button_Print_Top"
onclick="openPrintWindow()"><IMG
src="/intraworks/internet/frameless.nsf/v9906/IW/$File/Print.gif" border=0
alt="Print"></SPAN><BR><BR>
<DIV id="Contact_Links" class="Links"></DIV>

```

After the title [IMAGE] This option is often used to go to a new row/cell of a table.

```
<BR><BR>
```

After the end of the Page

```

</DIV>
</DIV>
<SPAN class="Bottom_Button_Back" onclick="history.go(-1)"><IMG
src="/intraworks/internet/frameless.nsf/v9906/IW/$File/Back.gif" border=0
alt="<< Back"></SPAN>
<SPAN id="CopyrightNotice" style="left:38%;"><CENTER>&copy; 1996 -
2005 Intraworks Solutions B.V.</CENTER></SPAN>
<SPAN class="Bottom_Button_Print" onclick="openPrintWindow()"><IMG
src="/intraworks/internet/frameless.nsf/v9906/IW/$File/Print.gif" border=0
alt="Print"></SPAN><BR><BR>
</DIV>

```

6.3 Determining the used layout definition

Layout definitions When debugging a Passage website, it is possible to determine which layout definition is used. Execute the following steps:

1. Load the page in the browser
2. View the source

3. Search for the text:

```
<!-- LookupKey= ?layoutdefinition?-->
```

Whereas ?layoutdefinition? stands for the used layout definition document

7 Web front end formatting issues

7.1 javaScript Library

The JavaScript library is used to store generic JavaScript routines. The advantage of this is that they are linked to the browser relatively, which means that they are included in the cache mechanisms. Obviously this creates more lightweight sites and pages, thus improving responsetimes and reducing data-traffic.

Key for this document	The key by which this JavaScript document is referenced.
Description	General description of the functions.
Content	The JavaScript routines being used.
Layout definition	The JavaScript routines being used obviously need to be included on the pages. The so-called binding of these pages is done in the Layout Definition (more relevant information can be found in that chapter).

Example usage for JavaScripts are:

- View formatting with custom sort routines
- Sitemap creation
- Menu control

7.2 Cascading Style Sheet (CSS) definition documents

The Cascading Style Sheet Definition library is used to store generic CSS Definitions. The advantage of this is that they are linked to the browser relatively, which means that they are included in the cache mechanisms.

CSS Definitions	Examples
HTML elements	
BODY	{font-family: Verdana; font-size: 75%; color: #6D6968; background-color: #EEEEEE; scrollbar-base-color: white; scrollbar-face-color: white; scrollbar-track-color: white; scrollbar-shadow-color: #396BA6; scrollbar-highlight-color: white; scrollbar-3dlight-color: #396BA6; scrollbar-darkshadow-color: white; scrollbar-arrow-color: #396BA6;}
BODY TABLE	{font-size: 100%;}
A	{text-decoration: none; color: #396BA6;}
A: hover	{color: black; text-decoration: underline;}

Many Examples are given on the Example page. Obviously these need tuning for your specific situation.

7.3 JavaScript views

In Passage 1.8 is the possibility to build up views in the browser. The Domino server generates JavaScript commands, which refer to JavaScript commands that are stored in a JavaScript library (linked to the view via the layout definition). For example each view entry triggers a vE function that appends the passed data to an array. In the 'after the body' HTML field of the layout a finish command in JavaScript is given, and the data is written to the browser.

JavaScript views available

The following views are currently available:

1. Library JS

2. What's new JS **Fallbacks:**

The source is no longer visible, previous and next buttons will no longer work.

Download:

JavaScript files that format views can be downloaded from our site www.passage.nl

Sample JS View:

```
var aViewEntries = new Array();
```

```
function TDB( className ) { document.write( "<TD class=\""+ className + "\">" ) }
```

```
function TDE() { document.write( "</TD>" ) }
```

```
function TRB( className ) { document.write( "<TR class=\""+ className + "\">" ) }
```

```
function TRE() { document.write( "</TR>" ) }
```

```
function openDoc( docID ) {
```

```
// Opens the document given by DocumentUniqueID in read mode
```

```

document.location = DBNameHTML + "/0/" + docID + "?OpenDocument";
}

function viewEntry( docTitle, docID, docDepartment, docPhoneNumber, docFaxNumber,
docEmail, docPDate ) {
// Object constructor
// Refer to this function to see which data is available in the view
this.docTitle = docTitle;
this.docID = docID;
this.docDepartment = docDepartment;
this.docPhoneNumber = docPhoneNumber;
this.docKeywords = docFaxNumber;
this.docEmail = docEmail;
this.docPDate = docPDate;
}

function vE( docTitle, docID, docDepartment, docPhoneNumber, docFaxNumber,
docEmail, docPDate ) {
// Create an object instance in the array
oViewEntry = new viewEntry( docTitle, docID, docDepartment, docPhoneNumber,
docFaxNumber, docEmail, docPDate );

if (aViewEntries.push )
pushed = aViewEntries.push( oViewEntry )
else
aViewEntries = aViewEntries.concat(oViewEntry )
}

function writeTable() {
// Write out the final HTML

// Number of columns to be used
var iColumns = 1;

// Writing script
var i = 0;
var className1;
var record
var iAViewEntriesDivider = aViewEntries.length / iColumns;
if( iAViewEntriesDivider == 0 ) iAViewEntriesDivider = 1;
if(aViewEntries.length==0) i = 1;

for( var iColumnNumber = 1; iColumnNumber<=iColumns;iColumnNumber++) {
TDB( 'TData'+iColumnNumber);
// document.write(iColumnNumber+'<BR>');

for( i = 0; iColumnNumber * iAViewEntriesDivider > i; i++ ) {

record=i+1;

if (i & 1)
className1 = "Odd"
else

```

```
className1 = "Even"
```

```
document.write( '<DIV class="' + className1 + '"><FONT class="TitleSmall"><A  
href="JavaScript:openDoc(\'" + aViewEntries[ i ].docID + '\')" class="Hyperlink">' +  
aViewEntries[ i ].docTitle + '</A></FONT><FONT class="Summary"><B>e-mail:</B> ' +  
aViewEntries[ i ].docEmail + '</FONT><IMG src="/icons/ecblank.gif" width=5  
height=5><FONT class="Summary"><B>Phone number:</B> ' + aViewEntries[ i  
.docPhoneNumber + '</FONT></DIV><IMG src="/icons/ecblank.gif" width=5  
height=5><HR>');
```

```
}  
TDE();
```

```
}
```

```
}
```

Please note:

JavaScript is a programming Language. You do need Programming Skills to modify JavaScript code. When using JavaScript views you do need extensive knowledge on JavaScript. Numerous resources are available on the Web.

The examples provided are as is and are no part of the Passage CMS product. Support on JavaScript can be given by Intraworks based on hourly rates.

8 Mail-In Filters

8.1 Introduction

Mail-In Filters are made to automate the storage of incoming e-mail by sender in the Passage database. Passage automatically creates a new library document for each incoming e-mail. For each sender you can define:

Library
keywords
workflow
status
audience
validity

Basics Document Categorization Notification Miscellaneous Settings	
Mail-In Filter	
Name	"Make a database mail-in resource and let me process the mail (I am SUPER)"
Order	"1000"

Filter Rules	
Match options	<input checked="" type="radio"/> Match any of the following <input type="radio"/> Match all of the following
the	Sender of the message
	Contains
More	

There are 2 types of mail-in, filtered mail-in and structured mail-import.

8.2 Defining Mail-In Filters

Configuration

Name Fill in the name of the mail-in filter
Order Order number, which one must be handled at first

Filter Rules It can consist out of more than one filter rule.
Fields that can identify the information source are:
- Sender
- Subject
- To fields

Document categorization The mail-in filter can automatically fill in several passage fields.

Audience Here you can specify the audience by clicking the names of Persons and Groups existing in the Address Book.

Information category Choose the Library Information Category you want this e-mail to be stored in

Keyword / Meta keywords Choose a keyword as defined in the keyword libraries document at Library level

Icons

Status Choose a icon from the image store "Passage icons"

Validity This will set the document into the chosen state.
The validity chosen here will occur in the library document in the

	bottom field
Notification	A Mail-In filter contains the functionality to send notifications. It can send a mail to an e-mail address. Also it can send notifications to the Next editor via Passage.
<i>E-mail notification</i>	Fill in the "Notification address" and set "notification by e-mail" to Yes .
<i>Next editor Passage</i>	Fill in the "Next editor"
Miscellaneous Settings	The name of the form or view to which the definition applies.

Always prepend the date to the title can make it easier to see what's new the current day in a view to the title

What should be done if a document has the same title and is in the same Library Information Category as an existing document?

The word 'slower' behind 'Keep existing document' and 'Delete existing document' means this task asks a bit more performance from the server compared to the 'Nothing' option.

Nothing' means: publish the document the way it is, so there will be two different documents with the same name in the same Library Information Category.

8.3 Structured mail import

Structured mail import is used in projects where the sender of the mail is under control of the Passage database owner. Using this functionality, the sender can be programmed in great detail on how the mail should be formatted, and where the e-mail is stored in the Passage database.

Items to be programmed in the mail are:

- Library category
- Keywords
- Validity
- Workflow status
- Sender id
- Current and Next editor
- Target body field (e.g. rich-text or text)
- Title
- Overwrite function
- Special fields

Please contact Intraworks for additional information on this functionality.

9 Document life cycle management

Passage has an agent that periodically searches the database for expired and revised documents. This agent generates an e-mail message to all editors in chief of the database. Please ask your system administrator to enable this agent.

10 Customisizing Passage code

Never touch the design of the Passage database without written confirmation of Intraworks. You are in violation of copyright laws modifying Passage code.

After authorisation it is possible to customise Passage, for example to:

- Store extra keywords on a document, and do some calculations
- Create custom search screens on the web Modifying Passage design will mean that automatic upgrade becomes problematic. To prevent this, custom code can be put into subforms loaded on the documents.
- Create related documents or input/order/... forms to have a content centric experience

Subforms are:

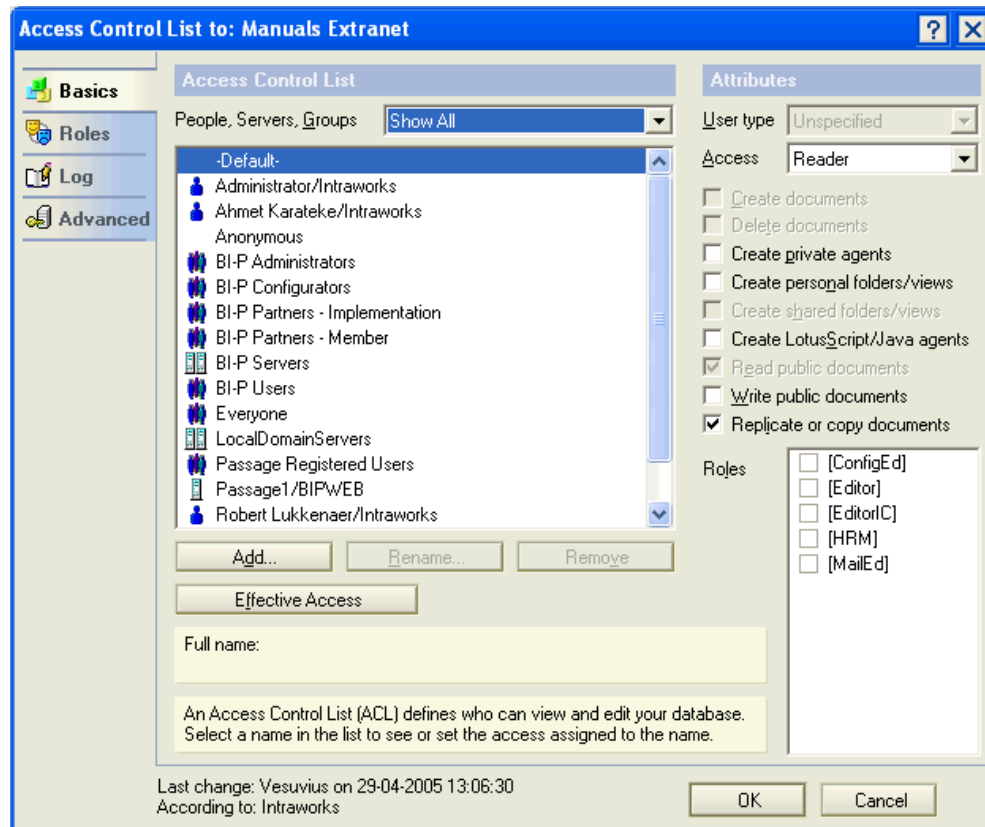
- s5000BBA1 Library
- s5001BBA1 Address book
- s5006BBA1 Brochures
- s5007BBA1 Events
- s5004BBA1 FAQs
- s5002BBA1 News
- More to come

10.1 Configuring the Database / Security

Passage workflows are configured using the ACL (access control list) of the database. The Administrator (database manager) assigns roles to employees in the ACL

In the ACL these roles should exactly be spelled as the below image. Notice the use of capitals! If the roles are not spelled correctly some of the Passage functions won't

work. The default access level for the Passage database in Notes is **Author**.



The **MAXIMUM** access level for the Passage database on the WEB **Author**.
The Configuration Editor (**ConfigEd**) has access to the Passage database configuration settings and the authority to change these. The access level for this role must be **Editor**

- Editor** Editors (**Editor**) can create documents and concept versions and forward these to the Editor in Chief.
- EditorIC** The Editor in Chief (**EditorIC**) can approve documents, publish them and remove them from the site.
- MaiEd** Mail Editors (**MaiEd**) are authorised to read incoming mail and to forward it to other Editors.
- HRM** Employees with the role of Human Resource Manager (**HRM**) are authorised to read job applications that come in and to process them.

For additional server settings, please refer to the Passage Administrators Manual.